

PROYECTO DE GRADO

MONITOREO Y DETECCION TEMPRANA DE INCIDENTES



INSTITUTO UNIVERSITARIO AERONAUTICO

INTEGRANTES:

GASTALDI MATIAS
PUSTILNIK EMILIANO

PROFESOR TUTOR:

BOGGIO ALEJANDRA

INDICE

1	INTRODUCCION.....	3
1.1	Fundamentación	3
1.1.1	Antecedentes	3
1.1.2	Situación Problemática	4
1.1.3	Objeto de Estudio y Campo de Acción.....	4
1.2	Hipótesis.....	4
1.3	Objetivos	5
1.3.1	Delimitación del Proyecto	5
1.4	Marco Contextual	6
1.4.1	Entorno del objeto de estudio	6
1.4.2	Relación tesista y objeto de estudio	6
2	MARCO TEORICO	7
2.1	Fundamentos de la alerta temprana	7
2.2	Sistemas de Monitoreo y Alertas Tempranas en Bases de Datos.....	9
2.3	Monitoreo Proactivo de Bases de Datos.....	10
2.4	Claves al monitorear una Base de Datos	11
2.5	Reingeniería de procesos.....	12
2.6	Aporte Práctico	13
3	DIAGNOSTICO Y ESTADO DEL ARTE	14
3.1	Diagnóstico.....	14
3.2	Antecedentes de proyectos similares.....	14
4	ANALISIS Y DISEÑO.....	16
4.1	Requerimientos.....	16
4.2	Arquitectura de la Solución.....	16
4.3	Planificación	19
4.4	Desarrollo de Indicadores	20
4.5	Análisis de las Fuentes de Datos	21
4.6	Creación del Modelo Físico	22

5	IMPLEMENTACION Y PRUEBA FUNCIONAL.....	24
5.1	Implementación General	24
5.2	Ejecución de pruebas funcionales reales.....	26
5.3	Análisis de salidas y conclusiones.	30
6	CONCLUSIONES.....	31
7	Anexo	34
7.1	Medición de tiempos	34
7.2	Scripts usados en la etapa de Pruebas Funcionales.....	38

1 INTRODUCCION

Trimix S.R.L. (en adelante Trimix) es una pequeña empresa que desea dejar de serlo y lograr un crecimiento sostenible. Para ser exitosa en esta transición, debe formalizar y sistematizar los procesos que hasta la fecha se vienen realizando con alta carga operativa y manual. El área Oracle Technology Services (OTS) tiene como principal objetivo que los sistemas de gestión de base de datos cumplan con:

- Optimización del rendimiento.
- Alta disponibilidad de los sistemas críticos.
- Recupero ante desastres.
- Acompañamiento y soporte.

La diferencia entre el éxito o fracaso de estos objetivos están directamente relacionados con la capacidad y velocidad de respuesta que el equipo pueda brindar. Se hace evidente, entonces, la necesidad de contar con información oportuna, precisa y confiable para brindar un servicio de excelencia, indispensable dada la criticidad de los servicios prestados por el área.

1.1 Fundamentación

1.1.1 Antecedentes

Del primer contacto con los responsables de la empresa, se pudo constatar que se trata de una Pyme de tipo familiar con 9 años de experiencia en el mercado. En general, ha experimentado un crecimiento sostenido desde sus inicios. Cada uno de los socios/gerentes es un verdadero especialista técnico del área que le toca gerenciar. Las diferentes áreas facturan autónomamente y tienen asociado su propio centro de costo, intentando de esta forma fomentar una sana competencia interna, trazando objetivos y metas de ingresos anuales.

La rentabilidad de cada área es, anualmente, puesta a consideración de la gerencia general de la empresa. Si existe algún área deficitaria, la misma se encarga de cubrir dicho déficit.

Del relevamiento realizado, surge que uno de los principales inconvenientes se presenta por la dificultad de mantener sueldos competitivos a sus colaboradores. En el mercado de Córdoba, operan numerosas empresas multinacionales dedicadas a la consultoría y *Software Factory* que trabajan con economía de escala. Dichas compañías, tienen a disposición gran diversidad de mercados globalizados en donde pueden obtener proyectos de alta rentabilidad.

Además, si bien los clientes extranjeros están familiarizados con el proceso inflacionario de Argentina, no pueden afrontar los incrementos anuales requeridos para readecuar los sueldos, restando paulatinamente competitividad a los profesionales involucrados en sus proyectos.

1.1.2 Situación Problemática

El área de OTS¹ administra sistemas de gestión de información de alta criticidad. Los errores o fallos en los sistemas operativos o bases de datos del cliente, pueden causar un grave perjuicio económico y/o un serio problema de imagen para con sus clientes finales.

En concreto, gran parte de la justificación y beneficio de contratar un servicio de soporte y mantenimiento viene dado por el grado y velocidad de repuesta brindada por los profesionales ante un eventual problema. No reaccionar a tiempo puede generar al cliente la sensación de que está pagando por un servicio ineficaz o con escaso valor agregado.

Para que el área sea rentable, no es posible disponer de un administrador de base de datos (DBA) por cada cliente. La empresa actualmente cuenta con seis DBA para satisfacer las necesidades de doce clientes que poseen diversos entornos, sistemas operativos y bases de datos. En conclusión, la realidad es que los profesionales están superados en número y responsabilidades.

1.1.3 Objeto de Estudio y Campo de Acción

Luego de lo expuesto previamente, consideramos que el desarrollo e implementación de un Sistema de Monitoreo y Alerta Temprana (SMAT) contribuirá efectivamente a disminuir la probabilidad de ocurrencia de fallos considerados críticos en los sistemas de entornos productivos. De esta forma, reduciremos los costos que los clientes deberían afrontar si su producción se ve obligada a parar por un imprevisto.

Un efectivo y completo SMAT tiene cuatro funciones principales:

- ✓ *Análisis de riesgos*: conlleva una sistemática recolección de datos para la detección de posibles riesgos y vulnerabilidades.
- ✓ *Monitoreo y alertas*: implica el estudio de los factores que indican que un fallo es inminente como así también los métodos utilizados para para detectar estos factores.
- ✓ *Comunicación*: es clave contar con la capacidad de notificar preventivamente la posibilidad de ocurrencia de un fallo a las personas involucradas para que puedan tomar las acciones pertinentes y así, mitigar o corregir la situación. Dicha notificación debe ser clara, precisa y estar dirigida a quien corresponde en tiempo y forma.
- ✓ *Capacidad de respuesta*: debe ser adecuada y requiere la construcción de un plan de contingencia ante un posible fallo. Dicho plan, tiene que detallar claramente las acciones a ejecutar y contar con suficiente evidencia de que funciona.

1.2 Hipótesis

Luego de analizar la situación problemática planteada por Trimix, consideramos que la solución óptima es implementar un Sistema de Monitoreo y Alerta Temprana mediante el cual un tablero de comandos

¹ Oracle Technology Services

reflejará aquellos indicadores asociados al entorno del sistema de información que sean considerados críticos. Los mismos deberán ser monitoreados periódicamente por los administradores y, en paralelo, el sistema deberá ser capaz de notificarles en el momento correcto la ocurrencia de una eventual situación no deseada.

Por otro lado, nuestra solución deberá ser lo suficiente inteligente como para proveer al administrador del sistema la retroalimentación adecuada para anticiparse un posible fallo y, de esta forma, pueda tener toda la información necesaria para tomar la decisión correcta a tiempo.

Nuestra propuesta es desarrollar un sistema que cumpla con estas características y que, además, almacene cada evento en una base de datos del tipo relacional. El tablero de comandos mostrará los indicadores en tiempo real. Un administrador, con conocimientos básicos en base de datos y sistemas operativos, podrá de manera rápida y sencilla monitorear todas las variables críticas donde los sistemas productivos están actuando.

Por último, es crucial que a partir de esta base de datos, creemos una nueva base de datos con información histórica. La misma estará orientada a consultas para la toma de decisiones, ya que al disponer de dicha información histórica y agregando análisis estadísticos y de correlación, se podrá rápidamente anticipar los fallos y generar nuevos puntos de mejora en los ambientes productivos de los clientes.

1.3 Objetivos

El Gerente de OTS manifestó la necesidad de realizar mejoras en el manejo de la información en los entornos de los clientes y que era imperioso idear un circuito que permitiese procesar datos críticos de la forma más eficiente posible, con el objetivo de mejorar la calidad del servicio prestado.

Este pedido nos motivó a analizar la situación del área y a pensar la forma de desarrollar un proyecto que además de subsanar el problema planteado, genere valor agregado al servicio prestado.

Estamos convencidos que el monitoreo para la detección temprana de incidentes beneficiará los procesos del área. Además, el diseño e implementación de un tablero de control con indicadores gráficos de contraste visual, posibilitará detectar eventuales inconvenientes e indagar a “n” niveles más bajos, hasta detectar el origen de la alerta generada.

1.3.1 Delimitación del Proyecto

De acuerdo a la hipótesis planteada y los objetivos mencionados, nuestro proyecto propone estudiar y desarrollar una plataforma en la que se encuentren agrupados por temática los principales indicadores de la base de datos bajo análisis y del sistema operativo en la que se encuentra y de esta forma hacer más eficiente el trabajo diario de un DBA.

El proyecto comienza con la identificación y captura de los requisitos y necesidades de quienes serán los futuros usuarios o consumidores de la información resultante de este trabajo. Se observará la forma

diaria de trabajo del equipo de DBA, identificando aquellas tareas que son manuales y repetitivas, como así también aquellas que son automáticas y/o sistémicas. Se releva también si existe algún proceso de trabajo que deba respetarse o la existencia de alguna norma de calidad a la cual atenerse. El resultado de esta etapa inicial será un listado con aquellos requerimientos que nuestro trabajo deberá cumplir para ser exitoso. Como en todo proyecto, habrá requerimientos críticos que no podrán faltar y otros considerados como requerimientos “deseados” o de baja importancia.

Este trabajo finalizará con la implementación de una herramienta que sirva de apoyo a las tareas cotidianas de análisis y monitoreo de los principales aspectos de un Sistema de Base de Datos como así también, aspectos de prevención de futuros errores o fallos.

Utilizaremos una metodología evolutiva e incremental basada en hitos en la que nuestro Cliente tiene una importante participación en el desarrollo y aceptación de este trabajo. Se harán presentaciones parciales cuando un hito es alcanzado y éste tendrá que ser validado y aprobado para continuar con la siguiente etapa.

Inicialmente este trabajo cubrirá la necesidad de información, monitoreo y principales alertas de un sistema de base de datos Oracle en versiones 9i o superiores, dejando para una versión futura el desarrollo de un Datawarehouse que permita realizar otro tipo de análisis más específico.

1.4 Marco Contextual

1.4.1 Entorno del objeto de estudio

La idea de este proyecto surge y es enaltecida por el contexto macroeconómico turbulento de nuestro país, donde los costos de importar productos tecnológicos o adquirir licencias de software son muy elevados para una Pyme que compite en un mercado relativamente chico. Trimix analizó la posibilidad de adquirir licencias de un enlatado con características similares a nuestra propuesta, pero llegó a la conclusión de que era imposible afrontar esa inversión por los costos asociados. Además, no tenían la certeza de que fuera a cubrir las expectativas.

1.4.2 Relación tesista y objeto de estudio

Uno de los factores determinantes a la hora de llevar a cabo este desarrollo, fue la necesidad de reducir costos en el contexto previamente mencionado sin dejar de brindar un servicio de alta calidad.

Los consultores de esta compañía poseen una vasta experiencia en la administración de bases de datos y sistemas operativos que sumada a la que nosotros como tesistas tenemos en el desarrollo de software, convierten a esta problemática y su resolución en algo sumamente desafiante. Luego de repasarlo, no dudamos en tomar esta idea para el desarrollo de nuestro trabajo de grado.

2 MARCO TEORICO

Un sistema de monitoreo y de alerta temprana de incidentes es un elemento importante en la reducción de riesgos ante un fallo del sistema. Intenta prevenir anomalías reduciendo de esta forma el impacto económico que los incidentes generan en el Cliente y el Proveedor del servicio.

Desde hace más de 10 años, se ha producido un gran interés en el desarrollo de sistemas de monitoreo y alerta temprana. Ellos pueden utilizarse para detectar una amplia gama de eventos, las posibilidades abarcan desde colisiones vehiculares y brotes de enfermedades hasta lanzamientos de misiles o catástrofes naturales.

Nosotros pretendemos aplicar los conceptos y principios de estos sistemas en la detección temprana de incidentes de Bases de Datos Oracle, aprovechando la gran cantidad de indicadores de rendimiento que este motor proporciona a sus usuarios.

Para ser eficaz, el diseño debe involucrar activamente tanto al DBA como al Cliente propietario de dicha base de datos (el dueño de la información).

Consideramos que, si bien nuestra idea está desarrollada y se aplica a la problemática planteada por una empresa en particular, nuestro sistema podrá ser implementado en cualquier entorno cuya base de datos sea Oracle en las versiones 9i en adelante. El esfuerzo de implementación del mismo se limita al relevamiento y posterior desarrollo de aquellas variables, métricas y eventos que sean considerados como relevantes y que requieren una medición y monitoreo.

2.1 Fundamentos de la alerta temprana

Un sistema de alerta temprana es el conjunto de políticas asociadas a tecnologías y procedimientos, diseñados para predecir y mitigar el daño de acontecimientos indeseados provocados por humanos u otros eventos. Un desastre ocurre sólo si el objeto o individuo expuesto a un peligro no puede hacer frente a sus efectos. Por otro lado, el peligro existe cuando hubo una vulnerabilidad no detectada a tiempo.

El objetivo es prevenir pérdidas o reducir el impacto económico y material de los incidentes. Para ser eficaces, los sistemas de alerta temprana deben involucrar activamente a las comunidades en situación de riesgo y generar una concienciación de los mismos, difundir eficazmente las alertas y las advertencias y garantizar un estado de preparación constante. Un sistema completo y eficaz de alerta temprana se apoya en cuatro funciones principales:

- ✓ **Conocimientos de los Riesgos:** se deberán, en esta etapa, recopilar sistemáticamente datos y realizar evaluaciones de riesgos. Se deberán responder las siguientes preguntas: ¿Se conocen bien los peligros y las vulnerabilidades? ¿Cuáles son los patrones y tendencias en estos factores? ¿Están ampliamente disponibles los mapas y datos de riesgo?

- ✓ Monitoreo y alertas: ¿Se están monitoreando los parámetros correctos? ¿Existe una base científica sólida para hacer pronósticos? ¿Se pueden generar advertencias precisas y oportunas?
- ✓ Difusión y comunicación: Comuniquen la información de riesgo y las alertas tempranas ¿Las advertencias llegan a todas las personas en riesgo? ¿Se entienden los riesgos y las advertencias? La información de advertencia es clara y utilizable.
- ✓ Capacidad de respuesta: Crear capacidades de respuesta a los responsables ¿Están los planes de respuesta actualizados y probados? ¿Se utilizan las capacidades y los conocimientos locales? ¿La gente está preparada y lista para reaccionar ante las advertencias?

Los buenos sistemas de alerta temprana tienen fuertes vínculos entre los cuatro elementos. Los principales actores involucrados con los diferentes elementos se reúnen periódicamente para asegurarse de que comprenden todos los demás componentes y lo que otras partes necesitan de ellos. Los escenarios de riesgo son construidos y revisados. Se acuerdan e implementan responsabilidades específicas a lo largo de la cadena. Se estudian los acontecimientos pasados y se realizan mejoras en el sistema de alerta temprana. Se acuerdan y publican los manuales y procedimientos. Se consulta a las comunidades y se difunde la información.



Merece especial atención mencionar que la Alertas deben estar categorizadas y que es necesario que se cuente con protocolos que indiquen qué actividades se deben ejecutar, qué procedimientos se deben implementar y cómo se debe dar un seguimiento al evento. A modo de ejemplo, se pueden clasificar del siguiente modo:

- ✓ Verde: Existen las condiciones generales para que se presente el fenómeno, se debe estar atento al comportamiento y evolución del fenómeno o evento monitoreado que las alertas que se continúen emitiendo
- ✓ Amarilla: Se están generando las condiciones específicas para un fenómeno potencialmente grave. Se inician los preparativos para ejecutar las acciones correspondientes, dirigidas a enfrentar el impacto del evento y sus consecuencias.
- ✓ Naranja: Se han concretado las condiciones necesarias para que se presente el fenómeno y sólo es cuestión de tiempo para que se manifieste.
- ✓ Rojo: Ya se manifestó el fenómeno y ha causado o está causando daños.

Para lograr la comunicación eficaz de las alertas los avisos deben ser cortos, simples y precisos; se debe proporcionar información oportuna sobre la amenaza, indicar acciones y medidas que deben tomarse para reducir las pérdidas y daños, explicar las consecuencias de no hacer caso a la alerta y contener verbos activos y repetir periódicamente información importante.

Fuentes: Conceptos extraído de la página oficial de la UNISDR (United Nations Office for Disaster Risk Reduction - Oficina dependiente de las Naciones Unidas)

- ✓ <http://www.unisdr.org/2006/ppew/whats-ew/basics-ew.htm>
- ✓ <http://www.unisdr.org/2006/ppew/info-resources/ewc3/checklist/English.pdf>

2.2 Sistemas de Monitoreo y Alertas Tempranas en Bases de Datos

Los sistemas de alerta temprana, como todo sistema de información, se define como:

"El conjunto formal de procesos que, operando sobre una colección de datos estructurada de acuerdo a las necesidades de la empresa, recopila, elabora y distribuyen selectivamente la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones necesarios para desempeñar funciones de negocio de la empresa de acuerdo con su estrategia" (1).

De acuerdo a la teoría general de sistemas, los componentes básicos de un sistema son: entradas, procesos y salidas. La fase de recopilación es cuando se receptan las entradas, que son estímulos que provienen del contexto donde opera el sistema (es decir, los llamados Inputs). Estos estímulos desatan una serie de acciones que responden a los procesos previamente establecidos y entregan una salida dirigida al mismo contexto como respuesta al estímulo que la originó.

Con estos conceptos se pretende destacar la importancia de controlar la información y cómo el uso de herramientas de gestión de información eficientes supone un avance importante. Un extracto de una nota obtenida de la consultora KPMG llamada "*Los datos son información La información es valor*" dice:

"La cantidad de información crece de manera explosiva, y el valor de los datos como activo de las organizaciones está ampliamente reconocido. Para que los usuarios obtengan el máximo rendimiento de sus enormes y complejos conjuntos de datos son necesarias herramientas que simplifiquen las tareas de

administrar los datos y de extraer información útil en el momento preciso. En caso contrario los datos pueden convertirse en una carga cuyo coste de adquisición y de gestión supere ampliamente el valor obtenido de ellos".

Quedando clara la importancia de las bases de datos, podemos definir éstas como un conjunto de datos pertenecientes al mismo contexto y almacenados para su posterior uso, dentro de un entorno digital gestionado mediante software y hardware. El software y hardware están en un proceso de cambio continuo, por lo que resulta importante saber en qué punto de desarrollo se encuentra cada tecnología para conocer qué nivel de beneficio se puede obtener de su uso. (2)

(1) *Estrategias y Sistemas de Información - Andreu, R, & Ricart, 1996*

(2) *Date. Introducción a los Sistemas de Bases de Datos.*

2.3 Monitoreo Proactivo de Bases de Datos

El monitoreo del rendimiento de la base de datos puede ser utilizados para mitigar problemas y ayudar a las organizaciones a ser más proactivas, de modo que puedan evitar problemas de desempeño e interrupciones. Incluso la base de datos mejor diseñada experimenta una degradación del rendimiento. Si los problemas de rendimiento no se corrigen rápidamente puede ser perjudicial para la rentabilidad de una empresa.

Debido a que el desempeño de los sistemas y aplicaciones de bases de datos puede verse afectado por múltiples factores, las herramientas que pueden encontrar y corregir las causas de los problemas de rendimiento de las bases de datos son vitales para las organizaciones que se basan en sistemas de gestión de bases de datos (Data Base Management System o DBMS).

Hay muchos tipos de problemas de rendimiento de base de datos, lo que a menudo hace que sea difícil localizar la causa de los problemas individuales. Es posible, por ejemplo, que las estructuras de la base de datos o el código de aplicación estén viciadas desde el principio. Las malas decisiones de diseño de base de datos y las sentencias SQL incorrectamente codificadas pueden resultar en un bajo rendimiento.

O puede ser que un sistema fuera bien diseñado inicialmente, pero con el tiempo los cambios causaron que el rendimiento empezara a degradarse. Más datos, más usuarios o diferentes patrones de acceso a los datos pueden ralentizar incluso las mejores aplicaciones de bases de datos. Incluso el mantenimiento de un DBMS (o la falta de mantenimiento regular de las bases de datos) puede causar que el rendimiento caiga considerablemente.

Los siguientes son tres indicadores importantes que podrían indicar problemas de rendimiento de base de datos:

- ✓ Aplicaciones cuya performance se ve degradada. La indicación más importante de problemas potenciales de rendimiento en la base de datos es cuando las cosas que solían correr rápido empiezan a correr más lento. Esto incluye los sistemas de procesamiento de transacciones en línea que usan los empleados o clientes, o los trabajos por lotes que procesan datos en grandes cantidades para tareas tales como el procesamiento de la nómina y los informes de fin de mes.

- ✓ Puede ser difícil monitorear las cargas de trabajo de procesamiento sin herramientas de gestión de rendimiento de bases de datos. En ese caso, los administradores de bases de datos (DBA) y los analistas de rendimiento tienen que recurrir a otros métodos para detectar problemas, en particular, las quejas de los usuarios finales acerca de cuestiones tales como que las pantallas de la aplicación tomen demasiado tiempo para cargar o que nada suceda durante mucho tiempo después de que la información se introduce en una aplicación.
- ✓ Interrupciones del sistema. Cuando un sistema está fuera de línea, el rendimiento de la base de datos, obviamente, está en su peor momento. Las interrupciones pueden ser causadas por problemas de la base de datos, tales como quedarse sin espacio de almacenamiento debido al incremento de los volúmenes de datos o por un recurso que no esté disponible, como un conjunto de datos, una partición o un paquete.
- ✓ La necesidad de actualizaciones de hardware frecuentes. Las organizaciones que están constantemente actualizando los servidores a modelos más grandes, con más memoria y almacenamiento, a menudo son candidatos para la optimización del rendimiento de base de datos. Optimizar los parámetros de la base de datos, sintonizar las sentencias SQL y reorganizar los objetos de base de datos puede ser mucho menos costoso que actualizar con frecuencia los costos de hardware y equipos.

Del otro lado, a veces se necesitan actualizaciones de hardware para resolver los problemas de rendimiento de la base de datos. Sin embargo, contando con la información necesaria y a tiempo, es posible mitigar los costos de actualización al localizar la causa del problema e identificar las medidas apropiadas para remediarlo. Por ejemplo, puede ser rentable agregar más memoria o implementar dispositivos de almacenamiento más rápidos para resolver los cuellos de botella I/O que afectan el rendimiento de una base de datos. Y hacerlo probablemente será más barato que sustituir todo un servidor.

2.4 Claves al monitorear una Base de Datos

Cuando surgen los problemas de rendimiento de la base de datos, es poco probable que su causa exacta sea inmediatamente evidente. Un DBA debe traducir las quejas vagas acerca de los problemas de los usuarios finales en temas específicos, relacionados con el rendimiento, que pueden causar los problemas que se describen. Esto puede ser un proceso difícil y propenso a errores, especialmente sin herramientas automatizadas para guiar al DBA.

La capacidad de recopilar las métricas sobre el uso de la base de datos e identificar los problemas específicos de la misma (cómo y cuándo se producen) es tal vez la capacidad más importante a la hora de monitorear una base de datos. Al enfrentar una queja de rendimiento, el DBA puede utilizar una herramienta para poner de relieve las condiciones críticas actuales y pasadas. En lugar de tener que buscar la causa raíz del problema de forma manual, el software puede examinar rápidamente la base de datos y diagnosticar posibles problemas.

Contando con la información correcta, se pueden establecer umbrales de rendimiento que, una vez disparados, alertan al DBA de un problema o activan un indicador en pantalla, en verde, amarillo o rojo. Además, los DBA pueden programar que los informes sobre el rendimiento de la base de datos se

ejecuten a intervalos regulares, en un esfuerzo por identificar los problemas que deben ser abordados. Las herramientas avanzadas pueden tanto identificar, como ayudar a solucionar los problemas.

Mediante el examen de las condiciones del umbral, el DBA puede utilizar la herramienta para identificar los cuellos de botella que causan la contención y luego remediar el problema rápidamente. Esta situación dista mucho respecto a la del DBA que no tiene las herramientas y permanece en la oscuridad con poca idea de lo que debe buscar.

En definitiva, contando con la información suficiente y oportuna, se podría reducir al mínimo la cantidad de tiempo necesario para resolver los problemas de rendimiento de base de datos y, a veces, incluso ayudar a las organizaciones a evitar algunos problemas de rendimiento e interrupciones completas. Por ejemplo, disponiendo de dicha información, las acciones de un DBA pasan de ser reactivas a ser proactivas acerca de la resolución de problemas de rendimiento. En lugar de esperar a que los síntomas de rendimiento mediocre sean notificados por los usuarios del sistema, los problemas pueden ser descubiertos y se puede tomar medidas correctivas, ya sea por el DBA o automáticamente por una herramienta, tal vez antes de que los usuarios sean incluso conscientes de un problema.

Fuente: <http://searchdatacenter.techtarget.com/es/cronica/Bases-de-datos-vitales-para-el-exito-de-las-empresas>

2.5 Reingeniería de procesos

Es importante reconocer que la reingeniería de procesos es un concepto aplicado globalmente y por el cual se tienen como expectativa mejorar la calidad y eficiencia de los procesos, para lograr iguales o mejores resultados con un uso óptimo de los recursos. Ésta técnica, que comenzó a implementarse en la década de los 90's, aplicada a los sistemas de información, encuentra también en este trabajo un ámbito de aplicación, ya que el nuevo sistema posibilitará la revisión y rediseño radical de los procesos de negocios que implican el monitoreo y resolución de los inconvenientes derivados del uso de bases de datos Oracle en la empresa.

Al no existir ningún proceso formal, ya sea comercial o procedimental que esté enmarcado en una norma de calidad (por ejemplo), el diseño de los nuevos procesos será moldeado por el comportamiento del flujo de información resultante del sistema de información a implementar. En el mismo sentido, tomando la totalidad de los clientes actuales, ninguno tiene definido formalmente el curso de acción, sino que es más dependiente del profesional actuante, por lo que la eficiencia del proceso queda relativizada a las capacidades individuales del conductor de dicho proceso.

Por lo tanto, al implementar el sistema y automatizar el proceso de comunicación y alertas del flujo de información, contribuimos a la eficiencia del mismo, estandarizando su calidad operativa.

Los beneficios tangibles e inmediatos de la aplicación de la reingeniería de procesos, considerando las 4 funciones fundamentales del sistema de alertas tempranas, son:

- ✓ Análisis de riesgos: Se podrá lograr una comunicación inmediata del riesgo inminente, las 24 horas del día y al momento de la ocurrencia del evento a informar.
- ✓ Monitoreo: Será realizado permanentemente mientras la base de datos se encuentre corriendo, sin la dependencia de una persona que ejecute los diagnósticos o los interprete.

- ✓ Alertas: Serán emitidas convenientemente según el grado de severidad de la misma, y/o la necesidad de la intervención de un profesional capacitado. En otras palabras, el nuevo proceso contará con cursos de acción para alertas que deben ser atendidas inmediatamente y con otras que puedan ser atendidas posteriormente ya que no poseen un impacto inmediato.
- ✓ Capacidad de Respuesta: Los nuevos procesos mejorarán la capacidad de respuesta, guiando el curso de acción a seguir posteriormente a recibir una alerta, con la capacidad de informar y tomar contacto con los involucrados en la resolución de forma inmediata, en cualquier momento del día, de ser necesario.

Por último, según Raymond L. Manganelli y Mark M. Klein, en su obra, “Cómo hacer reingeniería”- (1995), las herramientas deben lograr los siguientes beneficios:

- ✓ Mejorar la productividad.
- ✓ Buscar alcanzar mayores niveles de calidad.
- ✓ Eliminar trabajo aburrido y concentración en trabajo con valor agregado.

Tomando como eje los beneficios que deseamos alcanzar continuamos con el desarrollo del presente trabajo.

Fuente: Reengineering the Corporation (A manifesto for Business Revolution) - Michael Hammer y James Champy - 2005

2.6 Aporte Práctico

Este proyecto tiene como propósito desarrollar un Sistema de monitoreo y detección temprana de incidentes que permita, mediante la aplicación de un tablero de comando, explotar los indicadores de performance que las Bases de Datos Oracle proporcionan. El objetivo es anticiparse a los potenciales fallos de las mismas y tomar las acciones correctivas adecuadas. Además, y gracias a las automatizaciones, se pretende hacer más eficiente el tiempo de trabajo de cada DBA y permitir la administración de más de una base de datos en simultáneo (con el consecuente beneficio monetario).

La investigación es tratada como una propuesta no experimental, porque estudia los hechos tal como se originan en su contexto natural, sin alterar o influenciar ninguna de las variables. Presenta varios enfoques de acuerdo a sus características, logrando definirla según los siguientes tipos:

- Según el nivel de profundidad:
 - ✓ Investigación Descriptiva, ya que se busca describir e identificar informes de gestión, de modo de exponer la situación actual de los mismos.
 - ✓ Se considera de tipo evaluativo, ya que se busca establecer una propuesta de diseño de indicadores de gestión.
- Según la estrategia:
 - ✓ Investigación de campo, ya que se tiene contacto directo con el personal involucrado en la solución.

3 DIAGNOSTICO Y ESTADO DEL ARTE

3.1 Diagnóstico

En las reuniones iniciales el gerente del área de OTS nos comentó la situación por la que su área desde hace un tiempo considerado estaba pasando y que en conclusión necesitaba organizar su área haciendo foco en la repuesta al cliente. Para esta persona, el tiempo de respuesta de sus empleados con respecto a algún incidente (ya sea un error productivo, alguna mejora o un cambio necesario) podían mejorar drásticamente y no solo eso, sino que además él estaba confiado que muchos de los errores críticos podían evitarse en un gran porcentaje.

Luego nos comentó que existían sistemas informáticos dedicados a la prevención de situaciones críticas pero por lo que estuvo viendo, no existía nada relacionados con ambientes informáticos en donde tanto las base de datos como el sistema operativo donde residen las mimas puedan ser monitoreados de una manera rápida, sencilla, mostrando información resumida de aquellos indicadores considerados críticos en estos ambientes. Por supuesto que sin descuidar el costo de implementación sea acorde al presupuesto que una PYME maneja.

Hace aproximadamente 2 años atrás este gerente designó un grupo de trabajo para dedicarle tiempo de I+D a este proyecto, pero como muchas veces ocurre en una empresa donde la cantidad de empleados es limitada como así también el presupuesto que manejan, este no llego a buen puerto debido a que no se pudo dedicar el esfuerzo necesario debido a las tareas diarias que cada uno de estos empleados desarrollaba. Es así como nosotros tomamos esta problemática y comenzamos a investigar.

Las primeras investigaciones fueron con el objetivo de encontrar algún sistema informático con una idea similar, pero no encontramos información suficiente, así que el paso siguiente fue buscar sistemas informáticos que sean del mismo tipo que el que nosotros apuntamos. Luego concluir la investigación, encontramos varios sistemas que tratan problemáticas parecidas a la nuestra. De cada uno de estos fuimos tomando algunas ideas iniciales que nos ayudaron a plantear un ante-proyecto el cual fue presentado y aceptado por esta gerencia.

Estas ideas iniciales se fueron desarrollando y adaptando éstas y nuevas más que fueron surgiendo a la problemática planteada y donde los análisis de viabilidad fueron aceptadas por la gerencia ya que esto representa un gran oportunidad de mejora, dando solución al planteo inicial de organizar el área en pos a una mejora en los tiempos de respuesta y anticipación a situaciones críticas, mejorando de esta forma el servicio brindado.

3.2 Antecedentes de proyectos similares

Como se menciona anteriormente y de acuerdo a una investigación que se llevó a cabo para determinar si algún software existente en el mercado se pudiera adaptar a los requerimientos planteados, llegamos a la conclusión de que si bien existen una variedad de software que tratan la problemática de detección temprana de incidentes, no encontramos ninguno que se enfocara a ambientes informáticos.

Dimos con que lo poco que se pudo encontrar en el mercado respecto a este tipo de sistemas, no se adaptaban exactamente a lo que se estaba buscando y aquello que mejor podría cumplir con los objetivos planteados, eran excesivamente caros para el presupuesto que se manejaba.

Por otro lado, ninguno de los softwares analizados incluía la posibilidad de analizar información histórica y resumida que de soporte a la toma de decisión, ya que únicamente el análisis de la información era de manera “on-line”, es decir, lo que estaba ocurriendo en ese mismo momento. A nosotros, además poder ver lo que está ocurriendo en ese mismo momento, creíamos que almacenar y analizar la historia de los eventos sucedidos, podría ser de gran utilidad y agregar mayor valor a toda la información entregada por el sistema. De esta forma, podríamos detectar por ejemplo, cuellos de botellas que podrían ocurrirse en ciertos días y horarios y que podrían ralentizar estos ambientes informáticos debido a grandes tráficos de información. También podríamos detectar patrones en fallos cuando se realiza backups o además, por ejemplo, el motivo (con fecha y hora) de llenado inesperado de un disco de almacenamiento o de un tablespace de la base de datos, etc.

De esta forma tendríamos los 2 tipos de información mencionadas, es decir, la on-line y la histórica conviviendo y complementándose una con otra y agregando valor a la información entregada al usuario final.

4 ANALISIS Y DISEÑO

4.1 Requerimientos

Esta captura de requisitos tiene como propósito definir las especificaciones funcionales, no funcionales y de la solución para la implementación de un sistema WEB que permitirá administrar y consultar la información de aquellas variables asociadas al entorno del sistema de información que sean consideradas críticas y que deberán ser monitoreadas periódicamente, utilizado por los DBA, gerencia y usuarios finales.

El método usado para esta captura fue mediante entrevistas a los diferentes DBA que actualmente desarrollan esta actividad manualmente, como así también, al gerente del Área a los que los DBA pertenecen.

Se detalla a continuación aquellas funcionalidades relevadas y que el sistema debe tener para lograr los objetivos anteriormente planteados.



Plantilla Historias
Usuario -Trabajo de C

4.2 Arquitectura de la Solución

La solución fue pensada multi-cliente y multi-entorno, esto quiere decir que en una misma plataforma podremos monitorear todas aquellas bases de datos que un cliente requiera en sus respectivos entornos (Desarrollo, Pre-Producción, Producción, etc). Para cada base de datos a monitorear, se requiere que la herramienta STATPACK esté instalada y habilitada. Por lo tanto esto será un pre-requisito a la hora de implementar esta solución.

STATPACK es una herramienta que contiene un conjunto de *scripts* que permiten el recolectado y almacenamiento de datos asociados a la performance de una base de datos. Estos datos son almacenados en tablas Oracle. Cuando Statspack es instalado, el usuario (o esquema) PERFSTAT es creado automáticamente. PERFSTAT es el dueño de todos los objetos necesarios para que Statspack funcione correctamente.

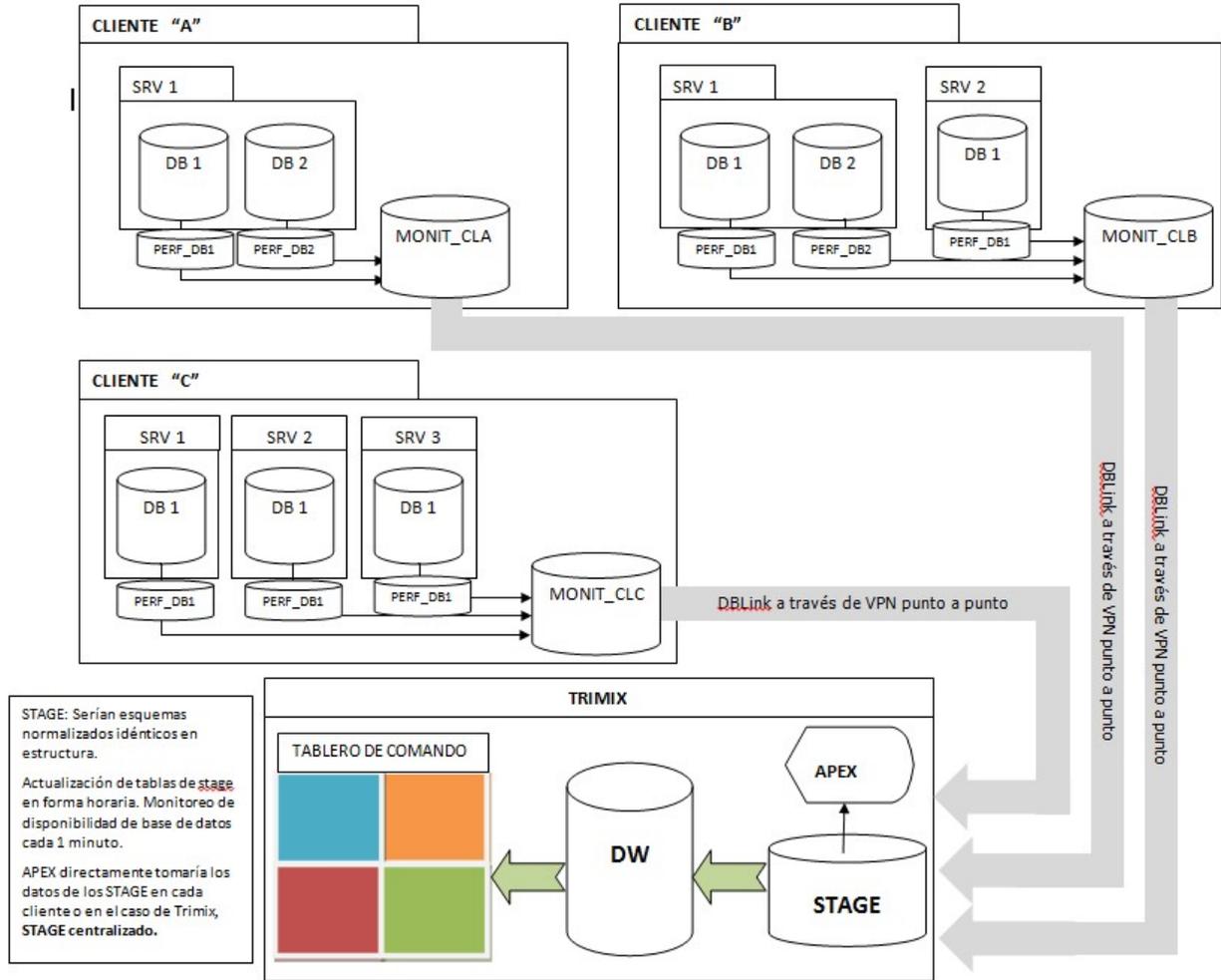
Adicionalmente, un segundo repositorio de información (o esquema) es requerido. A este lo denominamos TRX_MONITOREO y contendrá vistas y procedimientos almacenados sobre los objetos de PERFSTAT. En otras palabras, TRX_MONITOREO se establece como una capa superior sobre PERFSTAT. Por otro lado, este esquema almacenará la información necesaria acerca de los usuarios que acceden al sistema y los correspondientes permisos que previamente se le han otorgado.

Sobre TRX_MONITOREO se encuentra la herramienta Oracle Application Express (APEX). APEX es una plataforma de desarrollo de aplicaciones que consideramos mejor se adapta a la finalidad de este trabajo. Bien podría haberse utilizado cualquier otra plataforma (por ejemplo, Qlikview, Tableau, Oracle Business Intelligence, etc.) pero quisimos aprovechar la ventaja de que APEX está disponible con la instalación de la base de datos, además de su rápido aprendizaje y su simplicidad a la hora de desarrollar indicadores, reportes y tableros de comando visuales.

El monitoreo de múltiples usuarios en múltiples entornos será llevado a cabo mediante una base de datos que contendrá todas las conexiones necesarios para poder acceder a cada base de datos de forma remota de cada cliente. Este repositorio lo denominamos STAGE y en otras palabras, centraliza los denominados Database Link (o DBLink) necesarios para acceder a las mencionadas bases de datos que se encuentran de forma distribuida.

Para acceder al monitoreo de las bases de datos de un cliente en particular, el usuario deberá disponer de las credenciales necesarias para acceder a al sistema y visualizar información de ese cliente. Un super-usuario podrá ser configurado para tener acceso ilimitado a todos los clientes y sus bases de datos a monitorear. Un usuario con rol de Administrador deberá ser creado al momento de implementar este trabajo. Este usuario y todos aquellos con este rol asignado tendrán la opción de crear/editar/borrar un usuario como así también otorgar/quitar permisos a los mismos.

Por último quisimos en esta figura graficar la arquitectura descrita anteriormente. Se aclara que la temática de planteada de generar un Datawarehouse a partir de la información generada por este sistema queda para una versión posterior.



STAGE: Serían esquemas normalizados idénticos en estructura.

Actualización de tablas de stage en forma horaria. Monitoreo de disponibilidad de base de datos cada 1 minuto.

APEX directamente tomaría los datos de los STAGE en cada cliente o en el caso de Trimix, STAGE centralizado.

4.3 Planificación

El propósito de éste apartado es guiar la ejecución del proyecto y explicitar los resultados esperados. Esta sección abarca desde las actividades a realizar y los hitos o entregables hasta la definición de la agenda.

Supuestos: La propuesta de desarrollo del sistema en cuestión es la mejor alternativa considerada para atender a las necesidades de la empresa mencionadas anteriormente.

Inicialmente el proyecto apunta a la resolución de los problemas mencionados, en caso de que a lo largo del proceso de desarrollo apareciesen nuevas necesidades, estas serán tenidas en cuenta en la medida en que estén relacionadas con dicho proyecto.

La planificación de actividades que se presenta a continuación debería poder ser respetada, sin embargo, se reserva un margen de tolerancia en las fechas de entrega en caso de que surjan retrasos por imprevistos.

Entregables por etapas			Estimación
Tarea	Etapas	Hito	Tiempo de Desarrollo
Desarrollo de Indicadores	Análisis	Modelo_Logico_y_Req_Informacion.doc	15%
Análisis de Fuente de Datos	Análisis	Relevamiento_de_Fuentes.doc	5%
Creación Modelo Físico	Diseño	Modelo_Fisico.doc	20%
Migración de la Información	Desarrollo	Modelo Presentacion.doc	15%
Creación de la Metadata			
Generación de Reportes y tablero	Desarrollo	Reporte_y_Catalogo.doc	40%
Informe de Despliegue	Implementación	Informe_de_Despliegue.doc	5%

Se procede a continuación a describir cada una de las etapas que comprende este plan de actividades que se llevarán a cabo en este proyecto:

Desarrollo de Indicadores: En esta etapa se realizan todas las actividades de relevamiento de las necesidades de información del cliente con el objetivo de pasar de tener un listado de requerimientos a un listado detallado de indicadores. En esta etapa se desarrollan actividades de Definición de Variables de Análisis para pasar a la confección del correspondiente documento denominado Desarrollo de Indicadores, cuyo objetivo es detallar los indicadores claves de la problemática y sus reportes asociados. La solución resultante de este proceso de relevamiento debe brindar la información necesaria para entrar en acción y tomar decisiones operativas y estratégicas que aseguren el éxito del proyecto.

Análisis de las Fuentes de Datos: En este paso se identifica las diferentes fuentes de datos y dando importancia a la complejidad de acceder y obtener la información necesaria para dar inicio al desarrollo del proyecto. Este componente describe los principales conceptos del negocio, tomando como

referencia los puntos clave a tratar e identificando la Funcionalidad de los Reportes a diseñar, para ello es necesario consolidar la Información Fuente, para que se establezca un rápido acceso a la misma y que además sea útil para la aplicación de herramientas de análisis de la información, pudiendo con esto, generar tableros de control que permitan monitorear y medir distintos indicadores relevados.

Creación de un Modelo Físico: Se materializa en forma de Tablas, Relaciones, Vistas, Paquetes, Procedimientos, etc. el Modelo Lógico definido anteriormente. En esta etapa además se definen y crean los correspondientes procedimientos de ETL necesarios para llevar a cabo dicho modelo.

Migración de la Información y Creación de la Metadata: Una vez que se dispone del Modelo Físico se procede a la migración de la información. Esta tarea se realizará en dos etapas, primero se migra de la Fuente de Datos hacia un área intermedia y una vez que dicha información se ha validado se pasa a la segunda etapa donde se pasa hacia la Base de Datos final. Esta tarea se realiza mediante la ejecución de los correspondientes procesos ETL. Cuando se dispone de la información migrada y validada, se procede a la creación de la Metadata de presentación que sentará la base para la futura explotación de los indicadores relevados mediante su correspondiente tablero de comando.

Generación de Reportes: Una vez validada la Metadata se comienza con la generación de los distintos reportes acordados en la propuesta. Estos serán mostrados dentro de un tablero donde estos reportes estarán agrupados por áreas temáticas y mostrarán información gráfica y resumida de los indicadores relevados.

Despliegue: Por último, una vez alcanzada la última etapa del proyecto, se procede a documentar la información de despliegue. Esta debe contener toda la información necesaria para realizar una puesta operativa en ambiente destino, de todos los procesos y objetos construidos.

4.4 Desarrollo de Indicadores

Esta es la primera actividad de las etapas enumeradas y descriptas en la planificación. Como se mencionó, esta etapa partimos de la captura de requerimientos a un listado detallado con los indicadores claves de la problemática planteada, las fórmulas asociadas según corresponda, el reporte que lo contendrá y por último, una breve descripción técnica de cómo obtenerlos para los casos que sean necesarios esta explicación.

El objetivo de este punto es saber concretamente qué información es la que el cliente necesitar disponer, para luego tener una explicación acerca de cada uno de estos indicadores y una aproximación de la complejidad de obtención cada uno.

El entregable de esta etapa es un documento detallando lo mencionado anteriormente mostrando la siguiente información:

- ✓ **Indicador:** nombre del indicador
- ✓ **Requerimiento Asociado:** ID del requerimiento que da origen al indicador
- ✓ **Campos a mostrar:** listado de los campos a visualizar en el reporte o KPI

- ✓ **Filtros:** listado de los filtros (si es que existen) aplicados sobre las fuentes de datos que dan origen al indicador.
- ✓ **Solicitudes de información:** información requerida al usuario y que se aplica sobre el reporte o KPI. Ejemplo: fechas desde/hasta a mostrar.
- ✓ **Descripción:** breve descripción del indicador



INDICADORES.xlsx

4.5 Análisis de las Fuentes de Datos

En el punto anterior desarrollamos los indicadores involucrados de acuerdo al relevamiento donde por cada indicador destacamos su fuente asociada como así también los campos a extraer y los filtros, en caso de que apliquen, de cada uno.

El siguiente punto describirá la complejidad de acceder a los datos dichas fuentes. En nuestro caso, para resolver esta complejidad, llegamos a la conclusión que el uso de vistas SQL resuelve este problema.

Para llevar a cabo esta actividad, cada base de datos a analizar requiere tenga instalado y habilitado PERFSTAT (*). Por lo tanto, aquellos indicadores relacionados con estadísticas de performance de la base de datos se obtendrán de las siguientes tablas relacionadas a PERFSTAT:

- 1- PERFSTAT.STATS\$FILETATXS
- 2- PERFSTAT.STATS\$SNAPSHOT
- 3- PERFSTAT.STATS\$SQL_SUMMARY
- 4- PERFSTAT.STATS\$SYSSTAT
- 5- PERFSTAT.STATS\$SYSTEM_EVENT
- 6- PERFSTAT.STATS\$TEMPSTATXS

Por otro lado, tendremos información de monitoreo, tanto de la base de datos como del servidor que la contiene que se almacenarán en las denominadas TABLAS DE MONITOREO. Para obtener esta información se requerirá el desarrollo adicional (conjunto) de los scripts necesarios para poblar dichas tablas:

- 7- TRX_MONITOREO.TRX\$ADR_INCIDENT
- 8- TRX_MONITOREO.TRX\$ADR_PROBLEM
- 9- TRX_MONITOREO.TRX\$ALERTAS
- 10- TRX_MONITOREO.TRX\$BACKUP
- 11- TRX_MONITOREO.TRX\$IOSTAT
- 12- TRX_MONITOREO.TRX\$TABLAS
- 13- TRX_MONITOREO.TRX\$TABLESPACES

- 14- TRX_MONITOREO.TRX\$VMSTAT
- 15- TRX_MONITOREO.TRX\$TRANSACTION

Cada una de estas vistas estarán sobre las tablas de Perfstat y Monitoreo (explicados en el punto anterior) y por lo tanto, tendrán como fuente de datos ambos conjuntos de tablas. Esta capa tiene el objetivo de simplificar y abstraer el acceso a la información y de esta forma tener solo aquellos datos útiles a los fines de esta propuesta.

Sobre estas capas mencionadas, la herramienta de desarrollo Oracle Application Express (también conocida como APEX) (**) será montada y luego los diferentes reportes y tableros construidos con dicha herramienta serán los encargados de transformar los datos en información útil para la toma de decisiones.

El entregable de esta etapa es un documento detallando lo mencionado anteriormente mostrando la siguiente información:

- ✓ **Requerimiento Asociado:** ID del requerimiento que da origen al indicador
- ✓ **Indicador:** nombre del indicador
- ✓ **Repositorio Fuente:** tabla y/o vista de donde obtener la información
- ✓ **Vista Asociada:** consulta SQL sobre cada repositorio fuente que contiene la información, los campos y filtros necesarios para calcular cada indicador.
- ✓ **Vistas SQL (2da solapa):** detalla la consulta SQL que origina cada una de las vistas.



Indicadores y vistas.xlsx

(*) Es un paquete que Oracle introdujo desde la versión 8.1.7 que recopila la información de algunos de los aspectos más importantes en cuanto a rendimiento de la base de datos.

(**) Es una herramienta que permite diseñar, desarrollar e implementar aplicaciones, receptivas y basadas en bases de datos usando solo su navegador web.

4.6 Creación del Modelo Físico

A continuación se describen los principales conceptos del negocio de la compañía, tomando como referencia la Funcionalidad de los Reportes a diseñar, para ello es necesario consolidar la Información Fuente, para que se establezca un rápido acceso a la misma y que además sea útil para la aplicación de herramientas de análisis de la información, pudiendo con esto, generar tableros de control que permitan monitorear y medir distintos indicadores definidos por cada área que el proyecto abarque.

La siguiente información se presentará en 2 documentos adjuntos a continuación. Por un lado, el documento Modelo Físico, muestra cómo cada uno de los requerimientos captados en la sección Requerimientos es resuelto y de qué forma será mostrado.

El segundo documento llamado Funcionalidad de los Reportes, muestra cómo cada uno de estos indicadores estarán relacionados entre sí bajo un tablero de comandos y qué disposición tendrán cada tablero desarrollado.



Modelo Físico.xlsx

5 IMPLEMENTACION Y PRUEBA FUNCIONAL

5.1 Implementación General

Implementación en cada cliente: La arquitectura de la solución necesita la instalación de componentes del lado de cada cliente y servidor ORACLE a ser monitoreado. Por lo tanto, por cada servidor deberá completarse la siguiente secuencia:

1. Implementar el paquete de servicios STATSPACK², lo cual generará la base de datos PERFSTAT, que posee la información de las bases de datos del servidor, de la cual se nutren los procesos del sistema de monitoreo.
2. Creación del usuario TRX_MONITOREO y una serie de vistas personalizadas las cuales en su encabezado identifican el cliente, el servidor y la base de datos, además de la información propia de cada una.
3. Creación de un usuario para lograr el acceso remoto a través de un DBLink. Dicho usuario tendrá solamente acceso a la base de datos PERFSTAT y a las vistas personalizadas por motivos de seguridad y confidencialidad de información.
4. Instalación del software Cisco AnyConnect y configuración para generar la conexión VPN necesaria para lograr el vínculo seguro entre el cliente y Trimix.

Implementación en el servidor de Monitoreo:

1. Implementar el paquete de servicios STATSPACK, lo cual generará la base de datos PERFSTAT, que posee la información de las bases de datos del servidor, de la cual se nutren los procesos del sistema de monitoreo.
2. Creación del usuario TRX_MONITOREO y una serie de vistas personalizadas las cuales en su encabezado identifican el cliente, el servidor y la base de datos, además de la información propia de cada una
3. Instalación del software Cisco AnyConnect y configuración para generar la conexión VPN necesaria para lograr el vínculo seguro entre Trimix y cada uno de los clientes a través de adaptadores virtuales (uno por cada cliente).
4. Creación de una nueva entrada por cada cliente en el TNSNAMES.ORA (en Oracle_home/network/admin) donde haga referencia a la base de datos remota de cada cliente, utilizado el usuario de conexión y credenciales creadas en el punto "3)" de la implementación del lado del cliente.
5. Creación de los Jobs de ejecución de las consultas remotas que también se encargarán de almacenar en tablas en forma local e histórica, la información recolectada de los servidores de los clientes.

A continuación se detallan los objetos, usuarios, esquemas y permisos necesarios para la correcta implementación del sistema:

² Instalación y configuración del StartPack disponible en : <http://www.orasite.com/tunning-y-optimizacion/instalacion-y-configuracion-de-statspack>

- 1- Lo primero que haremos es conectarnos a la base de datos destino y debemos crear los siguientes esquemas: TRX_MONITOREO y PERFSTAT. Los scripts necesarios para esto son los siguientes:



Crear_User_TRX_MONITOREO.sql

- 2- Luego asociamos al esquema TRX_MONITOREO al esquema APEX correspondiente. Esto se podrá llevar a cabo desde la base de datos o por la aplicación APEX, mediante el usuario ADMIN INTERNAL.
- 3- Desde APEX, mediante ADMIN INTERNAL, se debe crear el workspace TRX_MONITOREO y asociarlo al esquema TRX_MONITOREO recientemente creado.
- 4- El siguiente paso es importar en la base de datos destino el esquema PERFSTAT sin registros (solo los objetos). Este import se obtiene exportando de la última versión disponible que tengamos en nuestro entorno de desarrollo.
- 5- Haremos lo mismo con el esquema TRX_MONITOREO. Finalizado este punto, los objetos necesarios de base de datos están instalados.



Import.sql

- 6- A continuación se crean los usuarios del sistema necesarios para la administración del mismo. Estos usuarios serán asociados al grupo de usuario Administrador. Este grupo no tiene restricciones y es el que podrá crear nuevos usuarios, asignarles permisos, etc.



Crear_Admin_Usr.sql

- 7- Desde APEX y conectado con el usuario TRX_MONITOREO, importaremos la aplicación y las imágenes.



f113 APEX 20150326.sql

- 8- En este punto ya tenemos la aplicación instalada y los esquemas de base de datos asociados al sistema. Lo que resta ahora es dar los permisos necesarios a los usuarios de base de datos, para realizar esto, tenemos 3 scripts donde la salida de cada uno de estos scripts, son los grants necesarios para que el sistema funcione:

Script 1:

```
SELECT 'grant '|| privilege ||' to '|| grantee || decode(admin_option, 'NO', ';',  
'YES', ' with admin option;')  
FROM DBA_SYS_PRIVS  
WHERE grantee IN ('TRX_MONITOREO', 'PERFSTAT');
```

Script 2:

```
SELECT 'grant '|| privilege ||' on '||owner||'.'||table_name||' to '|| grantee ||  
decode(grantable, 'NO', ';', 'YES', ' with grant option;')  
FROM DBA_TAB_PRIVS  
WHERE grantee IN ('TRX_MONITOREO', 'PERFSTAT');
```

Script 3:

```
SELECT 'grant '|| granted_role ||' to '|| grantee ||decode(admin_option, 'NO',  
';', 'YES', ' with admin option;')  
FROM DBA_ROLE_PRIVS  
WHERE grantee IN ('TRX_MONITOREO', 'PERFSTAT');
```

La salida de estos scripts será una serie de grants que se requieren para el correcto funcionamiento. Un ejemplo de la salida de estos scripts es la siguiente:



Grants.sql

5.2 Ejecución de pruebas funcionales reales.

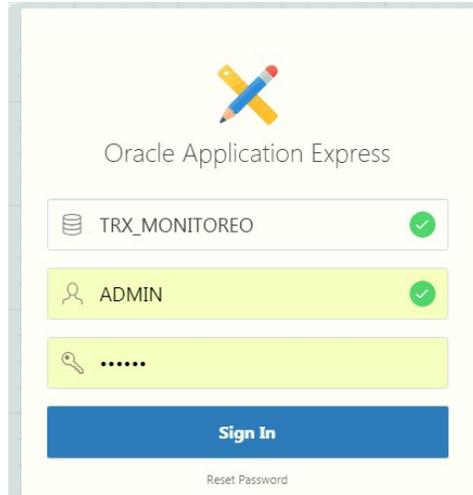
Una vez finalizada la implementación, se procede a validar la misma con el objetivo de verificar que todo haya quedado funcionando tal como se espera. La primer validación será ingresando al sistema recientemente instalado en el servidor destino y desde allí haremos un recorrido por todos los paneles del menú y de esta forma corroboraremos que todos los gráficos estén mostrando información, independientemente de que si esta información sea correcta o no, ya que a esta tarea la haremos en el próximo paso. En este solo verificaremos que la instalación de los objetos este correctamente hecha y que ningún grafico este dando error por algún objeto o permiso faltante.

- **Procedimiento de validación de implementación:**

1- Ingresamos a la aplicación Oracle APEX usando la url provista: <http://servidor:puerto/apex/>

Para nuestro caso es: <http://127.0.0.1:8080/apex/>

2- Nos logueamos al “Workspace” TRX_MONITOREO usando las credenciales provistas.



- 3- Una vez dentro del workspace, procedemos a levantar nuestra aplicación. Para esto, vamos al panel “Application Builder”

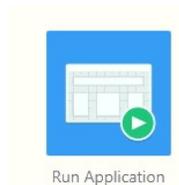


Application Builder

- 4- Y seleccionamos nuestra aplicación:

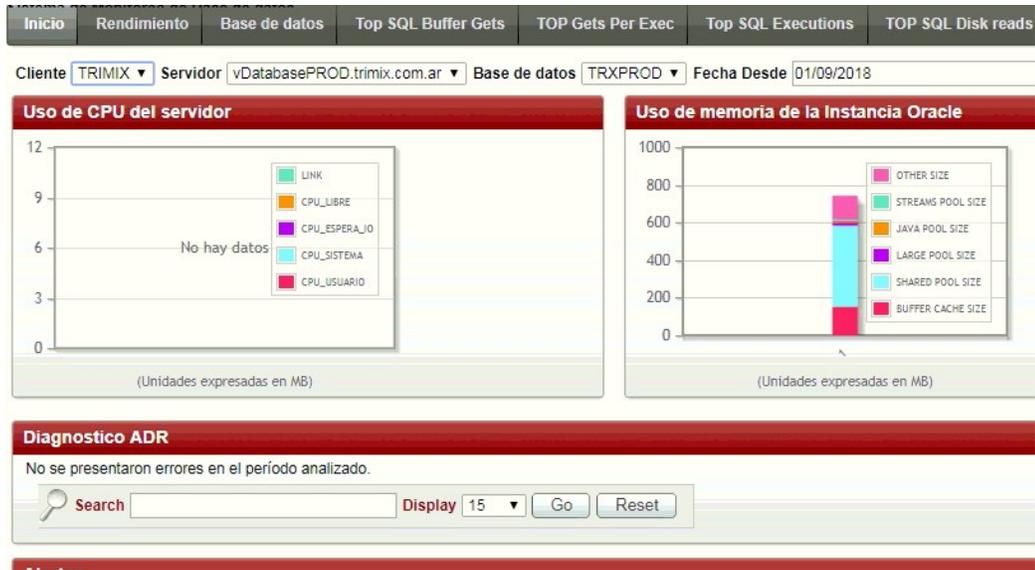


- 5- Y la levantamos seleccionando:



Run Application

- 6- Una vez que este corriendo, procedemos a hacer un recorrido por cada panel. Se prueban que cada filtro de cada reporte funcione correctamente y que ningún grafico muestre algún tipo de error.



Lo siguiente que haremos es verificar la información que cada indicador este mostrando sea correcta. Para llevar a cabo esta tarea, hemos creado una serie de scitps SQL donde cada uno de ellos tiene la consulta apropiada para validar cada uno de los gráficos mostrados.

El procedimiento de validación es el siguiente: Tenemos 26 scripts numerados del 01 al 26 donde cada uno recibe ciertos parámetros (por ejemplo, Nombre Cliente, Servidor, Fecha Desde/Hasta, etc.). Se toma el primer script en el orden establecido, se lo coloca en un editor SQL y se procede a ejecutarlo. El script solicitará que se ingrese el o los parámetros mencionados. Una vez ingresados y obtenido el resultado, se compara los dichos valores entregados por la consulta SQL contra lo mostrado en el correspondiente gráfico.

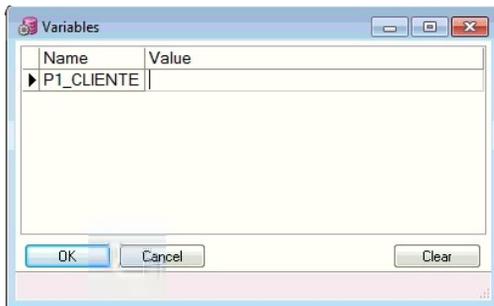
A modo de ejemplo, tomamos un indicador y su script de validación asociado y procedemos de la siguiente manera:

- 1- Supongamos que vamos a validar los valores arrojados por el primer grafico mostrado en el panel Inicio. Para esto, tomaremos el primer script llamado 01_Home_Uso_CPU_Servidor.sql y lo corremos en un editor SQL.

```

SQL Output Statistics
define P1_CLIENTE varchar2(50);
define P1_SERVIDOR varchar2(50);
begin
  SELECT *
  FROM ( SELECT cliente
         ,servidor
         ,cpu_usuario
         ,cpu_sistema
         ,cpu_espera_io
         ,cpu_libre
         FROM perfstat.v_cpu
        WHERE dim_tiempos = TO_CHAR (SYSDATE, 'yyyymmdd')
          AND cliente = &P1_CLIENTE
          AND servidor = :P1_SERVIDOR
        ORDER BY fecha DESC)
  WHERE ROWNUM = 1;
end;
    
```

- 2- Una ventana solicitando el ingreso de los valores de las variables necesarias y declaradas se mostrará para cada variable declarada:



- 3- El resultado arrojado por esta consulta será la siguiente:

	CLIENTE	SERVIDOR	CPU_USUARIO	CPU_SISTEMA	CPU_ESPERA_IO	CPU_LIBRE
▶ 1	TRIMIX	vDatabasePROD.trimix.com.ar	51	49	0	0

- 4- Estos valores los comparamos contra lo que muestra el grafico llamado "Uso de CPU del Servidor" usando, por supuesto, los mismos parámetros que los ingresados en la consulta SQL:



- 5- Comparamos ambos resultados: Vemos que en el gráfico, para los valores ingresados, tenemos que el valor para CPU_USUARIO es de algo más de 51 Mb y para CPU_SISTEMA es de 49. Estos

son los mismos valores arrojados por el SQL, por lo tanto este grafico está mostrando información correcta.

- 6- Este procedimiento se repite para diferentes combinaciones de valores de variables hasta validar que todas las pruebas fueron satisfactorias para el grafico en cuestión. Se procede a realizar lo mismo con cada uno de los restantes indicadores.

En el anexo [Scripts usados en la etapa de Pruebas Funcionales](#) se encuentran la totalidad de los scripts usados para esta validación.

5.3 Análisis de salidas y conclusiones.

Las pruebas realizadas, tal como se describe, fueron sobre la implementación de los objetos, verificando de esta manera, que ningún objeto y/o permiso este faltando. Por otro lado, se hicieron pruebas sobre la información mostrada, es decir, se toman los valores arrojados por los reportes y se los compara contra la información devuelta por el script correspondiente. Ambos procedimientos fueron descritos en el punto anterior.

Se aclara que no se realizaron pruebas sobre los datos almacenados en cada tabla usada por este sistema. Esto es debido a que dichos datos son de prueba y no representan ninguna realidad contra que compararse. Por lo tanto, desde este punto de vista, se toman estos como válidos y se parte de esta premisa para continuar las correspondientes validaciones mencionadas.

Para la totalidad de las pruebas realizadas mediante los procedimientos descritos, los resultados entregados fueron los esperados.

6 CONCLUSIONES

El objetivo fundamental de esta tesis era abordar la encrucijada en la que se encontraba el equipo de trabajo OTS (Oracle Technology Services) de Trimix. El área tiene como principal propósito la administración y gestión de las bases de datos de sus clientes y debe velar siempre por el óptimo rendimiento de las mismas, una alta disponibilidad de los sistemas críticos, el recupero ante desastres y el acompañamiento y soporte al cliente, entre otros.

Los problemas a los que la empresa se enfrentaba eran:

- ✓ Recursos escasos: insuficientes administradores de base de datos (DBAs) que, quizás por falta de herramientas, ejecutaban de modo frecuente tareas repetitivas, con alta carga manual y operativa que no generan mucho valor agregado.
- ✓ Calidad del servicio: para poder obtener participación en el mercado actual tan competitivo, deben brindar un excelente servicio, reducir los tiempos de respuesta a los clientes (situación más que complicada dada la cantidad de clientes por cada DBA), anticiparse a las fallas y contar con información integra, oportuna y confiable.

Para poder desarrollar una solución satisfactoria, en primera instancia realizamos un relevamiento de las tareas ejecutadas diariamente por el equipo. Se identificaron los requerimientos, necesidades y oportunidades de mejora. Rápidamente entendimos que la clave era automatizar ciertas tareas para lograr el ahorro de tiempo y dinero que la empresa perseguía y hacer la operatoria escalable para incorporar nuevos clientes. Se optó por desarrollar e implementar un sistema de monitoreo y alerta temprana (SMAT), el cual recopilara la información centralizadamente y permitiese mediante tableros de comandos no solo el monitoreo de la base de datos en tiempo real, sino también prevenir acontecimientos no deseados o la pronta respuesta, en caso de ser inevitables.

El campo de aplicación de un SMAT es muy amplio y versátil, pero es indiscutible que deben existir cuatro funciones imprescindibles (y muy bien coordinadas) para su correcto funcionamiento: Conocimiento de los Riesgos; Monitoreo y alertas; Difusión y comunicación y Capacidad de respuesta.

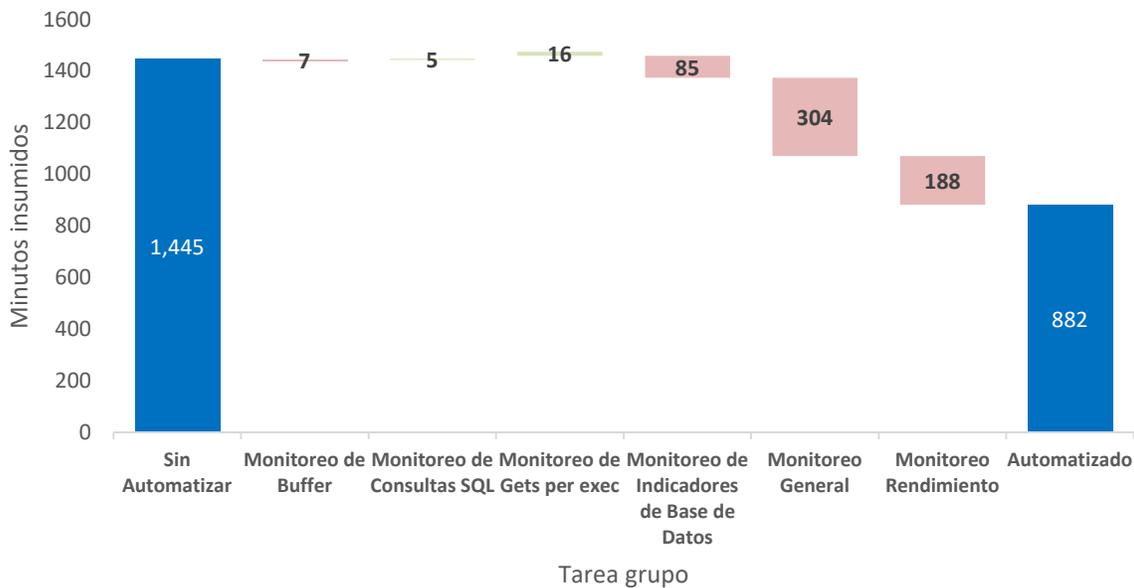
De esta forma, el aporte principal de este trabajo consiste en el análisis, diseño e implementación de un sistema de identificación y captura automática de aquellos aspectos claves de las bases de datos que hacen a su rendimiento. Debe mostrar las principales alertas suscitadas en un momento dado mientras agrupa los principales indicadores por tema, permitiendo una mejor comparación entre ellos y, en consecuencia, un mejor análisis para la detección de problemas producidos o anticiparse a potenciales problemas.

Si bien la información mostrada por este sistema requiere conocimientos un tanto avanzados en tareas de administración de bases de datos Oracle, esto fue pensando justamente así. El desarrollo del sistema

fue enteramente analizado y diseñado conjuntamente con un equipo de DBA, los cuales, en reiteradas reuniones, nos indicaban sus necesidades de información hasta llegar a una versión final que es la que estamos presentamos en este trabajo. Con esto queremos destacar que la información expresada y las interpretaciones a las que este equipo de DBA llega, quizás no sean intuitivas o de fácil interpretación para un usuario sin estos conocimientos.

Según los objetivos planteados al inicio de este proyecto, la empresa expresó conformidad con la solución, destacando cómo ahora tareas rutinarias se automatizan en el proceso diario de monitoreo e hicieron especial mención a la drástica mejora en el análisis de información, ahorrando tiempo y disminuyendo la posibilidad de cometer errores por la ejecución manual de scripts.

No conformes todavía con la aprobación y expresa conformidad del gerente de DBA y su equipo, creímos necesario ir un paso más adelante y medir ahora los nuevos tiempos insumidos por tareas de monitoreo comparado con los anteriores tiempos (ver [Anexo "Medición de Tiempos"](#)). Este informe nos arroja que los tiempos mejoran alrededor de un 39%, lo cual implica una importante reducción en los tiempos de tareas cotidianas y muchas veces rutinarias. En el siguiente gráfico podemos observar con mejor detalle:



El ahorro de tiempos se observa principalmente en las tareas de Monitoreo general (54%), Monitoreo de rendimiento (33%) y Monitoreo de indicadores de Bases de datos (15%), siendo los demás ahorros despreciables para este fin. La reducción en tiempos conseguida y lograda, posibilita que los DBAs realicen nuevas tareas, administren más bases de datos o sumen nuevos clientes.

Finalmente pensamos que una futura línea de investigación de este trabajo sería la creación y explotación de un Datawarehouse. Este contendrá la información histórica almacenada por este sistema

y permitirá un análisis más amplio y desde otras perspectivas con el objetivo de mostrar información resumida de todos los clientes para la toma de decisiones a nivel gerencial.

Con esta nueva información obtenida, un gerente podrá saber rápidamente si por ejemplo uno de sus clientes necesita más horas de consultoría debido a la gran cantidad de incidentes producidos en un tiempo determinado, por lo que podría decidir de ofrecer a este cliente un nuevo recurso. Podría también medirse la calidad del servicio prestado a nivel global, sabiendo cuánto es el tiempo que las bases de datos estuvieron sin servicio comparado a cuánto estuvieron funcionando correctamente. Con esto el gerente podría saber cuán eficiente son sus consultores, decidiendo por ejemplo, dedicar más horas a capacitación y/o investigación de nuevas tecnologías para hacer el trabajo más eficiente.

Estos son solo algunos ejemplos del potencial y el valor agregado que una base de datos como ésta podría aportar a una empresa. No solo puede mejorar el trabajo de un consultor o el área donde este se desempeña, sino también sentar las bases para un mejor y respaldado proceso de toma de decisiones a nivel empresarial.

De acuerdo a las conclusiones obtenidas por el análisis de este tipo de información podrían plantearse procesos estándares de trabajo en el área, con la consecuente mejora del servicio prestado a sus clientes. Además de las ventajas ya mencionadas en la sección [Reingeniería de procesos](#), consideramos que operar bajo estándares de trabajo proporciona un salto en la calidad del servicio prestado.

Si a la eficiencia en el uso del tiempo de los consultores le sumamos la estandarización de procesos, es inevitable pensar que gracias al SMAT estamos sentando los precedentes necesarios para hacer escalable la operatoria del área OTS

7 Anexo

7.1 Medición de tiempos

El siguiente anexo tiene como objetivo la medición de los tiempos, luego de la implementación del Sistema de Monitoreo y Alertas, en las tareas cotidianas de análisis y monitoreo por parte del equipo de DBA de la empresa Trimix SRL en las respectivas bases de datos de sus clientes. Luego esto es comparando esto con los tiempos insumidos previos a este sistema.

Para realizar esto, se tomó una muestra de 3 personas con una antigüedad de al menos 5 años trabajando como DBA, en el transcurso de 3 días laborales en la que únicamente se midieron de forma estimativa los tiempos insumidos en tareas de monitoreo y análisis de la información.

Cabe aclarar que estas mediciones son estimaciones dado que estas fueron hechas por las mismas personas que realizan las tareas y puede que carezcan de una precisión exacta, pero para los fines de este trabajo creemos que son más que suficientes.

Para el proceso de medición se desarrolló una aplicación en la que primero se parametrizan las tareas a medir y la base de datos relacionada a esas tareas, luego la persona que realiza las tareas, debe iniciar un *timer* que correrá en modo *background*. Una vez finalizada o pausada la tarea medida, el timer es manualmente parado/pausado y el tiempo transcurrido se imputa a la tarea seleccionada. De esta forma, automáticamente se irán sumando los tiempos por tarea y grupo de tareas. La misma tarea se podrá pausar y reiniciar las veces que sea necesario. También se pueden crear “sesiones de trabajo” para el caso de medir la misma base de datos en diferentes momentos.

Al finalizar este proceso, tendremos un listado de tareas monitoreadas por grupo de tareas realizadas en un período dado por un consultor a una base de datos en particular, tanto para el monitoreo realizado previo y post a la implementación de nuestro sistema.

Vale aclarar que el criterio usado por cada medición de tiempo consumido por una tarea es el siguiente:

<i>Sistema Implementado</i>	<i>Inicia</i>	<i>Cuando el consultor ingresa al sistema y navega hasta el indicar deseado.</i>
	<i>Finaliza</i>	<i>Cuando el consultor obtiene una conclusión luego de analizar el o los indicadores deseados</i>
<i>Tareas Manuales (Sin sistema)</i>	<i>Inicia</i>	<i>Cuando el consultor desea genera información para analizar un indicador. Para realizar esta Tarea, el consultor debe primero buscar en su repositorio de scripts aquel que desee ejecutar, En el caso de que exista, si no, generarlo. Luego, en un editor SQL o consola, ejecutarlo y Esperar el resultado. A partir de ese momento comienza recién empieza a monitorear y Analizar un indicador.</i>
	<i>Finaliza</i>	<i>Cuando el consultor obtiene una conclusión luego de analizar el o los indicadores deseados</i>

Se usó la siguiente plantilla para recoger los datos necesarios:

Sistema de Monitoreo y Alertas - Medición de tiempos insumidos por tareas							
Base de Datos ID	Consultor ID	Fecha Medición	Tarea Grupo	Tarea Detalle	Tiempo Insumido (en minutos)	Anotaciones	Tiempo Total por Grupo (en minutos)
			Monitoreo General	Revisión consumo CPU Revisión Uso memoria de la instancia Revisión de backups hechos Revisión de errores ADR Revisión de alertas generadas			
			Monitoreo Rendimiento	Análisis consumo CPU Análisis consumo CPU (Valores Maximos) Análisis Memoria Libre Análisis de Parseo (hardware vs software) Análisis de Logical Análisis de Physical Análisis de I/O			
			Monitoreo de Indicadores de Base de Datos	Listado de principales de Redo Logs Monitoreo de Crecimiento de la BD Monitoreo de Crecimiento de la DB Area Monitoreo de Tablas mas grandes y su evolución Monitoreo de Espacio Ocupado por Datafiles			
			Monitoreo de Buffer	Ranking de consultas SQL resueltas por el Buffer			
			Monitoreo de Consultas SQL	Ranking de consultas SQL que mas tomaron tiempo en ser resueltas por la BD			

A continuación se describe cada columna mostrada por esta plantilla:

- ✓ **Base de Datos ID:** muestra un identificador por base de datos analizada.
- ✓ **Consultor ID:** muestra un identificador por cada consultor que realizó la medición. Cabe aclarar que no nos interesa identificar la persona que lleva a cabo esta tarea con nombre y apellido, por lo tanto mantenemos el anonimato de la misma y sólo mostraremos un identificador único por cada una.
- ✓ **Fecha Medición:** muestra la fecha de cuándo se realiza la misma.
- ✓ **Tarea Grupo:** muestra una agrupación superior de las tareas o sub-tareas realizadas.
- ✓ **Tarea Detalle:** muestra las tareas a medir por grupo de tareas.
- ✓ **Tiempo Insumido:** Tiempo consumido por un consultor en una tarea determinada. Este tiempo es medido y reflejado en la plantilla por la misma persona que la realiza.
- ✓ **Tiempo Total por Grupo:** muestra el tiempo insumido por grupo de tareas.

El siguiente cuadro resume el alcance de las mediciones tomadas:

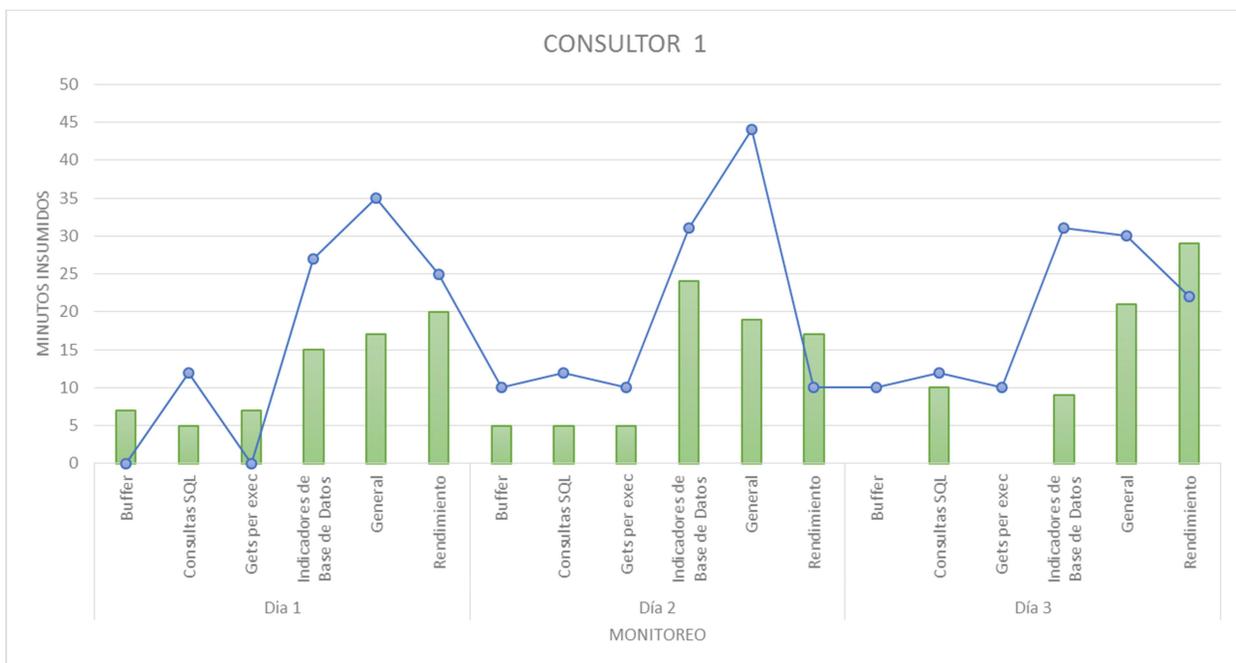
Sistema Implementado	Inicia	Cuando el consultor ingresa al sistema y navega hasta el indicador deseado
	Finaliza	Cuando el consultor obtiene una conclusión luego de analizar el o los indicadores deseados
Tareas Manuales (Sin sistema)	Inicia	Cuando el consultor desea genera información para analizar un indicador
	Finaliza	Cuando el consultor obtiene una conclusión luego de analizar el o los indicadores deseados

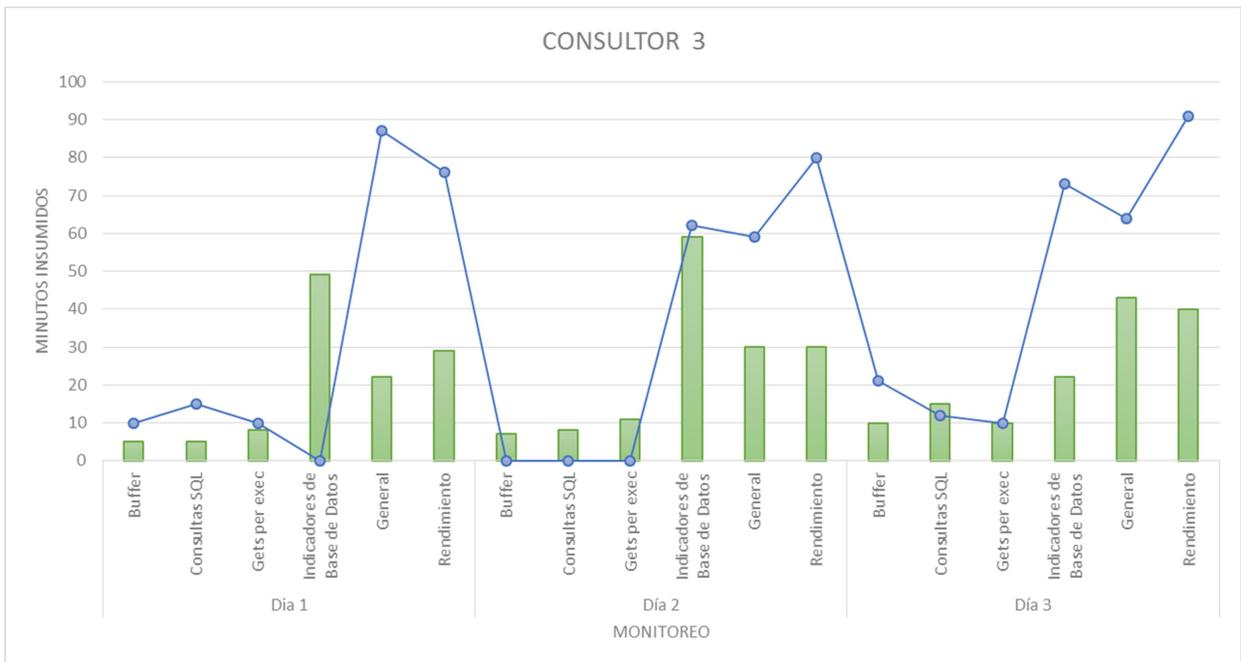
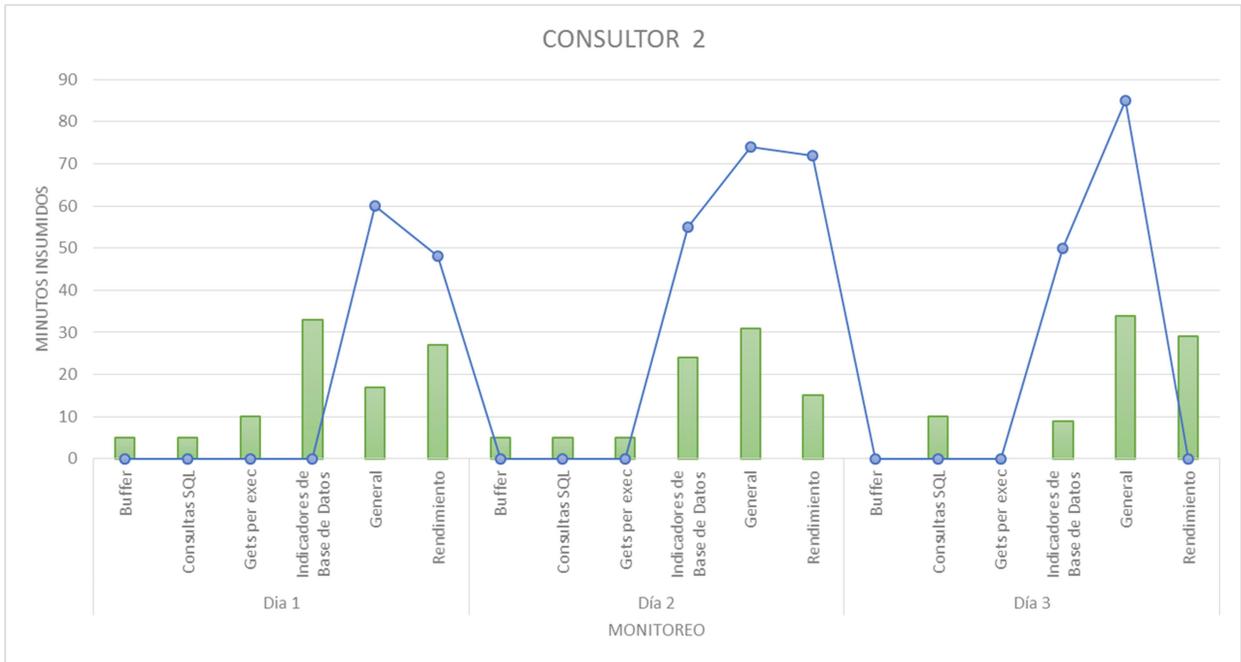
Los archivos con los datos recolectados son los siguientes:

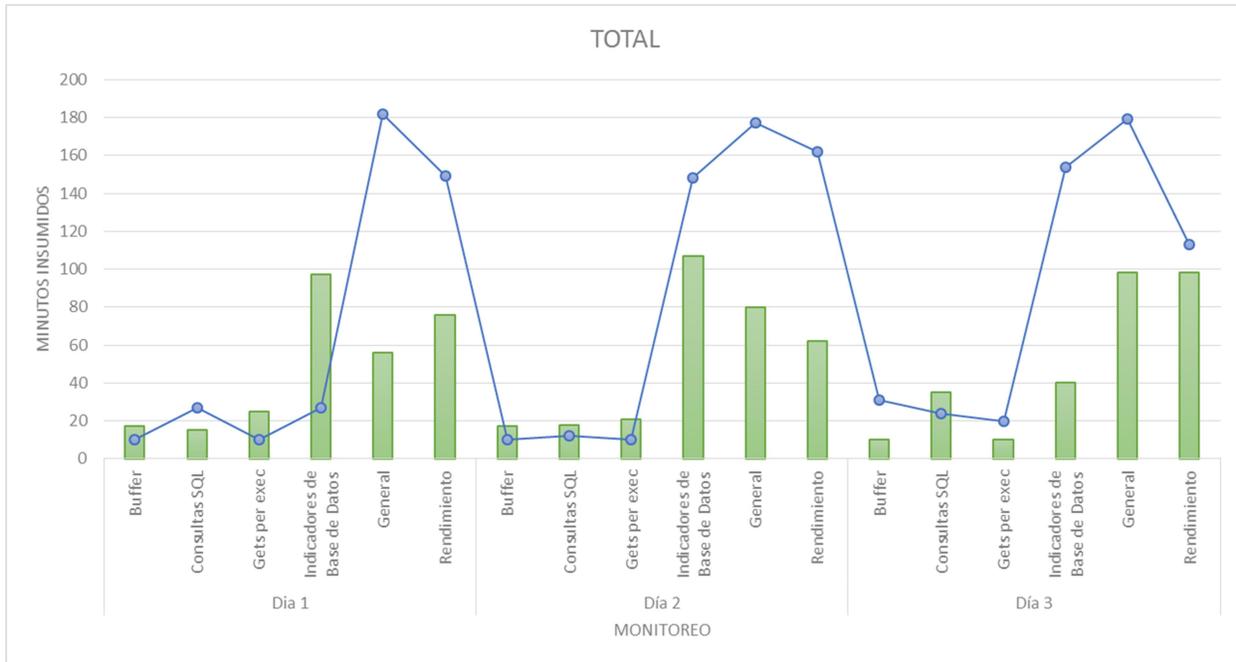
Consultor ID	Plantilla
C001	 MT_C001_Data.xlsx
C002	 MT_C002_Data.xlsx
C003	 MT_C003_Data.xlsx

Una vez recolectada los datos necesarios, procedemos a generar conclusiones a partir de los mismos. Para esto, comparamos los tiempos insumidos por cada tarea sin usar el sistema y luego comparamos los tiempos usando el sistema.

Gráficamente, al separar los resultados por cada consultor obtenemos:

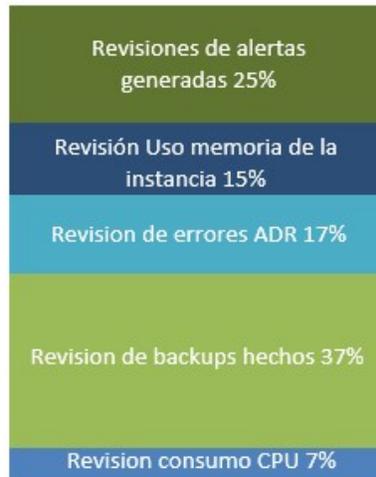






Luego de ver el evolutivo individual y colectivo, es evidente la mejora en el manejo del tiempo invertido en el Monitoreo general principalmente con la siguiente proporción de ahorros:

Monitoreo general



7.2 Scripts usados en la etapa de Pruebas Funcionales

A continuación se listan los scripts usados para validar cada gráfico mostrado por el sistema, usando el procedimiento descrito arriba:

Orden	Panel	Grafico	Script
1	Inicio	Uso CPU del Servidor	 01_Home_Uso_CPU_Servidor.sql
2	Inicio	Uso Memoria Instancia Oracle	 02_Home_Uso_Memoria_Instance_oracle.sql
3	Inicio	Resultado de Backup	 03_Home_Resultados_de_Backup.sql
4	Inicio	Diagnóstico ADR	 04_Home_Diagnostico_ADR.sql
5	Inicio	Alertas	 05_Home_Alertas.sql
6	Rendimiento	Rendimiento CPU Promedio	 06_Rendimiento_Cpu Promedio.sql
7	Rendimiento	Rendimiento CPU Valores Máximos	 07_Rendimiento_CPU Valores Maximos.sql
8	Rendimiento	Rendimiento CPU Valores Máximos Hora	 08_Rendimiento_CPU Valores Maximos Hora.sql
9	Rendimiento	Memoria Libre	 09_Rendimiento_Memoria_Libre.sql
10	Rendimiento	Parseo Hard vs Soft	 10_Rendimiento_Parseo_Hard_Soft.sql
11	Rendimiento	Logical Reads vs Writes	 11_Rendimiento_Logical_Reads_Writes.sql
12	Rendimiento	Physical Read vs Writes	 12_Rendimiento_Physical_Reads_Writes.sql
13	Base de Datos	Actividad de la Base de Datos	 13_IO_Activity_Database.sql

14	Base de Datos	Info General	 14_DB_Info_General.sql
15	Base de Datos	Tamaño DB	 15_DB_Tamano_DB.sql
16	Base de Datos	Tamaño DB Area	 16_DB_Tamano_DB_Area.sql
17	Base de Datos	Tablas más grandes	 17_DB_Tablas_mas_grandes.sql
18	Base de Datos	Transactions	 18_DB_Transactions.sql
19	Base de Datos	Espacio ocupado por Datafiles	 19_DB_Espacio_Ocupado_Datafiles.sql
20	Top SQL Buffer Gets	Top SQL Buffer Gets	 20_TOP_SQL_Buffer_Gets.sql
21	Top Gets Per Exec	Top Gets Per Exec	 21_TOP_SQL_Gets_per_Exec.sql
22	Top SQL Executions	Top SQL Executions	 22_TOP_SQL_Executions.sql
23	Top SQL Disc Reads	Top SQL Disc Reads	 23_TOP_SQL_Disk_Reads.sql
24	En Línea	Sesiones Activas	 24_EnLinea_Sesiones_Activas.sql
25	En Línea	Hit Cache Ratio	 25_EnLinea_Hit_Cache_Ratio.sql
26	En Línea	Sesiones en Espera	 26_EnLinea_SessionWaits.sql